Amendments to the Specification:

Please replace the paragraph beginning at page 13, line 6, with the following amended paragraph:

Base architecture 110 includes those components that function to connect the storage components to Java application 102. Base architecture 110 includes application-server-Java-database-connectivity services 112, LDAP Java Software Development Kit ("SDK") 114, and other application server services 116. A SDK provides APIs, programming framework and the like for ~~delvelopers~~ developers to use. LDAP Java SDK 114 provides the means to access the data in LDAP repository 106. Other application server services 116 may include logging, TX, EJB, and the like. Java-database-connectivity ("JDBC") services 112 ~~provides~~ provide an interface with database 104. LDAP Java SDK 114 provides an interface with LDAP repository 106.

A1

Please replace the paragraph beginning at page 13, line 15, with the following amended paragraph:

Base architecture 110 also includes persistent object framework ("POF") 120. POF 120 provides a uniform, simple application program interface for creating, updating, deleting and querying the persistently stored objects. POF 120 may provide transparent mapping of objects between memory, and database 104 and LDAP repository 106. POF 120 may provide features such as object relationships, concurrency protection, ~~cacheing~~ caching, deferred writing, and the like.

A2

Please replace the paragraph beginning at page 39, line 5, with the following amended paragraph:

Step [[308]] 304 executes by accessing persistent object information. This step also may be known as "introspection." POF 120 may access the persistent object for information on its type, attributes, relationships, and properties. The

A3

A3

following example pseudo-code discloses how to access a persistent object:

Please replace the paragraph beginning at page 56, line 14, with the following amended paragraph:

Thus, the steps that access the database are limited to performing the lazy load and searching by filter. These steps are executed after the cache is searched, which reduces the number of database accesses. Further, once an object is fetched from the database, [[DOF]] POF 120 caches the object so [[tat]] that the following query may retrieve the object from the cache directly without going back to the database.

A4

Please replace the paragraph beginning at page 56, line 19, with the following amended paragraph:

Fig. 6 depicts a flowchart for managing objects through relationships in accordance with an embodiment of the present invention. Fig. 6 includes searching, or retrieving, objects according to their relationships, as disclosed in step 522 of Fig. 5. Step 522, however, is not limited to the steps disclosed by Fig. 6. As noted above, a relationship expresses a link between two persistent objects. Step 600 executes by receiving an action to be performed via an object's relationship. Step 602 executes by retrieving related objects. Preferably, POF 120 retrieves objects in response to a search request from an application. POF 120 may retrieve objects according to multiple or singular cardinality relationship roles. The different methods of retrieving objects according to relationships may be disclosed according to the following examples. For multiple cardinality relationship roles, a variable named "buyer" that references an object of type "Company" or "Person" and a variable named "Order" that references an object of type "Order" may be given. The orders may be retrieved according the following example pseudo-code:

A5

Please replace the paragraph beginning at page 59, line 13, with the following amended paragraph:

Step 608 executes by removing relationships of an object. The different methods for ~~removng~~ removing relationships may be disclosed according to the following examples. For multiple cardinality relationship roles, a variable named "buyer" that references an object of type "Company" or "Person" and a variable named "order" that references an object of type "Order" may be given. The relationship between the "buyer" and the "order" may be removed according to the following pseudo-code:

A6

Please replace the paragraph beginning at page 60, line 17, with the following amended paragraph:

In this example, the "buyer_id" attribute defined in the "Order" object type is the sole "referencing attribute" and the "id" attribute defined in the "Person" object type is the sole "referenced attribute." This relationship may be known as a "foreign key" relationship. Relationships that are added, removed or changed may be made persistent when the transaction is committed or when the flush method is invoked.

A7